Welcome Home.

nyu_netid

Connect

password

Fall 2000

enter

Access and use of this computer system by anyone other than as permitted by NYU are strictly prohibited by NYU and by law and may subject an unauthorized user, including unauthorized employees, to criminal and civil penalties as well as NYU-initiated disciplinary proceedings. The use of this system is routinely monitored and recorded, and anyone accessing this system consents to such monitoring and recording.

about help login

Connect

Information Technology at NYU

Volume 11, Number 1 Fall 2000
Connect: Information Technology at NYU is edited and published by New York University's Information Technology Services (ITS). Its scope includes information about computing, networking and telecommunications across NYU's various schools, departments and administrative units, as well as developments in information technology outside the University.

Copies of Connect are available at the ITS Innovation Center, the ITS computer labs, the NYU Information Center and most graduate school offices. Copies are mailed to full-time University faculty, staff and researchers, based on mailing lists administered by Human Resources Division.

If you are a full-time faculty member and do not receive a copy, please notify your dean's office; full-time staff should notify their human resources representative. If you are not among these groups but would like a free subscription, send e-mail to its.connect@nyu.edu.

You can also read Connect online, through NYU Web, at the URL www.nyu.edu/its/connect.

We welcome your comments about the articles in this issue, as well as suggestions for future issues. Contributions are invited for consideration by the editor.

Opinions expressed in the articles in this publication are those of the authors and not necessarily those of Information Technology Services or of New York University.
Contents

NYUHome

Welcome Home!  
By David Ackerman
3
Building a Better Portal  
By David Ackerman
Building a Better Portal  
By David Ackerman
Tuning in to NYUHome’s Channels  
By Drew Hahn
5
NYU’s Portal Engine  
By Jonathan Vafai
6
The Foundation of NYUHome  
By David Ackerman
7
NYUHome: A System of Systems  
By Gary Chapman
8

Academics

Collecting Data Over the Web  
Data Entry for SPSS  
By Frank LoPresti
9
New Software Available through the Social Science Stats/GIS Group  
By Frank LoPresti
10
New Data Editor in SPSS Version 10.0  
By Frank LoPresti
11

Libraries

Webcasting in the New Millenium  
By Robin Schanzenbach, Gloria Rohmann, José Calero and Toni Urbano
12

Connectivity

New Uptown Connections  
Wireless Networking at the IFA  
By Yonatin Radzin
14

There’s No Place Like Home

NYUHome on Campus

By now, I’m sure you’ve heard about NYUHome, the new service that NYU is offering to our internet account holders. NYUHome is a portal — a door to many web-based services and tools, including e-mail. It lets you access your messages using either a web browser or the desktop e-mail program you’re already used to, such as Eudora or Outlook Express.

Having custom-built NYUHome for the NYU community, in collaboration with the Office of Electronic Publications, we here at Information Technology Services wanted to show it off a little. So we’ve dedicated most of this issue to exploring the many aspects of NYUHome.

Many people on campus have already been issued a new NYUHome account, and others have had their old is* accounts migrated over to the new platform. For those of you who will soon be migrated, but have yet to see what all the excitement is about, I hope this issue answers some of your questions. For more information, you can also check out http://home.nyu.edu/about/.

As we say hello to our new Home on the Internet, I also want to say goodbye. This will be my last issue as editor of Connect. It’s been a great three years, and I’ve truly enjoyed working with all of you. Thanks to all my readers for your feedback and praise. And to all my contributors, thanks for coming through on deadline most of the time!

— Joan Charlotte Matelli
Editor
YUHome is a centralized access point for University information, collaboration, interaction and communication. It gives individual students, faculty and staff members — and perhaps one day, alumni and prospective students — access to pertinent electronic information and services at NYU. The information and its presentation are custom-designed to fit each individual, and each individual can adapt NYUHome to fit his or her needs. Thus, NYUHome provides a valuable service to individuals associated with the University by consolidating the plethora of information that inundates all of us.

NYUHome is a type of software service commonly referred to as an Enterprise Information Portal (EIP). EIPs are also known as corporate portals, business portals, or simply portals. They are applications that provide a web-based, individually cus-

tomizable, consolidated view of the online information and services required by each individual associated with an organization. In NYUHome, information and services are contained in channels (see Drew Hahn’s article on page 6).

Perhaps most importantly, NYUHome will build community for the University by facilitating communication and enlarging the opportunities for information flow within the community. No longer will an individual need to seek out vital information; it will come to him. This will quickly become a very powerful communication mechanism at the University.

Reliability and performance are the key principals of the architecture of NYUHome. We assumed that as usage of electronic resources increases at the University, the NYU community will be less accepting of any significant periods of inaccessibility or poor response. Our current e-mail systems (the is* accounts) were built back when few people relied on e-mail and the Internet for their education or work. Today, many individuals cannot tolerate e-mail or web outages or delays for any significant length of time.

The hardware and system software that supports NYUHome are among the most advanced and powerful available today. And thanks to a gift valued in excess of $1 million from Sun Microsystems, NYU was able to acquire a new multiprocessor platform for the Shared Services Environment (SSE) that supports e-mail and web pages. Failover software is built into the SSE architecture, meaning that other processors are configured to take over if one processor fails (see diagram on page 10).

NYUHome is architected from the ground up to be extremely reliable. The multiprocessor platform, a Sun E10000, is formidable. This model contains four processor nodes (like four separate computers).

One node, devoted to e-mail, is as powerful as the nine machines that currently...
deliver NYU e-mail! And that’s only part of the overall processing power of the machine, which leaves capacity available for web services. With our e-mail and web access volume more than doubling every year, our Shared Services Environment can be scaled up to accommodate this growth.

The performance of the software is also of paramount importance. The multi-threaded architecture takes advantage of our multiprocessors. Extensive caching with a high hit ratio minimizes lookups. Of course, and perhaps most importantly, NYUHome is easy to use, and its attractive design makes for an enjoyable experience.

NYUHome is based on an object-oriented modular design. The software, called ISng (Information Services next generation — see Jonathan Vafai’s article on page 7), offers excellent functionality, including a complete language set that makes it possible to incorporate other information services.

In addition, NYUHome is flexible and scalable. Modifications and additions, including development of modules by autonomous teams, are relatively easy to incorporate. A preferences database, dependent on interfaces with other University systems (such as HR and SIS) for updates and triggers, also stores individuals’ choices, enabling a unique environment for each person (see Gary Chapman’s article on page 12).

NYUHome is not a stand-alone system, and over the next several years, we will continue to integrate it with other University services and systems.

---

**Building a Better Portal**

David Ackerman
david.ackerman@nyu.edu

When we were designing NYUHome, we were faced with the choice of buying an Enterprise Information Portal or building our own. This was not an all-or-nothing decision, however. Vendors have offered to host or give us portals, or to sell us the software for building a portal infrastructure. The list of products and companies we considered while assembling our EIP is large. But in each case, we would have needed to integrate NYU systems into the portal. Competing portals offer features or services of interest to NYU, so whichever portal NYU was to adopt would have to incorporate common portal features with existing NYU components.

A number of vendors offered us “free” portals. Of course, the vendor always has a cost recovery model. With one model, we would have had to agree to show ads on every portal page, in exchange for free portal infrastructure software. Another model offered free software in exchange for target marketing to NYU individuals. Although a “free” offer for portal software sounds good, there are three important fallacies in the arrangement.

**Fallacy:**

The most costly part of a portal is its software (which is what’s usually offered free in these deals).

**Fact:**

Most of a portal’s cost is in computing and networking infrastructure for the portal; staff to assemble, maintain and evolve the portal; and integration of the University’s existing information services into the portal.

**Fallacy:**

Giving away a part of the computer screen to outside advertisers costs nothing and will be well tolerated by the NYU community.

**Fact:**

The ad space in the “free” portals with advertising is quite valuable to NYU — more valuable, potentially, than any other
NYU space, real or electronic. We would be severely hampered in our competition for the eyes of the NYU community if we adopted “free” portal software.

Fallacy:
Free portal X will build community for NYU. (This is a popular vendor claim.)

Fact:
Free portal X is designed by Company X to build community and revenues for Company X.

NYU will need to continue to assess the desirability of acquiring external portal software on an ongoing basis. However, several factors led us to conclude that it is premature to buy a third-party product and make it our standard at this time:

• The EIP market is young and changing rapidly. There are no portal standards per se, other than the well-known Internet and Web standards. In addition, major software companies have announced their intention of entering the EIP market.

• Many of the companies with offerings are startups, and are not likely to survive the next few years. Some startups have teamed with more established companies in order to gain access to the educational market. Teaming with an established company merely creates the illusion of stability. Even established companies have immature products that may not survive.

• It’s not logical to pick a single solution at the exclusion of others. A number of the portals we have investigated have features of interest for NYU to adopt. Thus, why should we choose any single one, and miss out on features that another offers? We will want to pick and choose the best features from each, which we can only do by developing our own portal.

• The fallacy of picking a single solution is compounded by the fact that it is not logical to have a number of portal environments, each with fewer people. However, different schools, for example, could still individually brand NYUHome for their own smaller communities.

It is also often suggested that we have more than one portal. But offering multiple NYU portal products would dilute our goal of building NYU community. It would be far less desirable to have a number of portal environments, each with fewer people. However, different schools, for example, could still individually brand NYUHome for their own smaller communities.

In summary, having an NYU portal is fundamentally about building community, and at NYU, it is about doing so in a diffuse environment that doesn’t automatically lend itself to community-building. Anyone promoting the adoption of free portal software should understand the associated fallacies. Many of the claims made by vendors — in particular, that their product can take every piece of content we have and present it the way we want — are not credible.

In the context of NYUHome, there is no buy versus build dichotomy. Even as we buy (acquire), we will have to build. And as we build, we’ll often do so using purchased components, even for the infrastructure underlying NYUHome. It is premature to cede development of NYUHome to a vendor, even one offering a free portal with claims that it will do everything we want. We may choose to use third parties to help integrate existing services and systems; we will certainly be using a “best of breed” approach in implementing specific portal functions.

NYUHome is a gateway into the information world for members of the NYU community. As this world continues to rapidly evolve and transform the intellectual and work lives of us all, it will be necessary to continually evolve and renew the portal. Using our current approach, this will be attainable.
Tuning in to NYUHome's Channels

Drew Hahn
drew.hahn@nyu.edu

Channels, individual containers of information, are the logical organization structure of NYUHome. Most university web sites are created and designed according to their department, thus reflecting the physical structure of the University. In NYUHome, however, channels are given names that describe general types of information and may contain content from any area of the University, as long as it is specific to that keyword. This way of organizing university information represents a vital paradigm shift and creates vastly more efficient information searches.

As NYUHome matures and more channels are created, the emphasis will be on carefully choosing names for channels, grouping similar content, and making it easier to perform routine tasks on the data behind those channels.

Imagine reducing the number of clicks and form information submitted to retrieve information. Books from Bobst could be renewed with two clicks, access to Albert could be one simple click, your NYUCard balance could be displayed every time you log in. This is the future of personalized information with NYUHome.

To make the channels more useful, the data within them is tailored according to an individual's role at the University. A role is simply a way of defining a person's various relationships with NYU. For instance, Jane Doe could be a full-time staff member in the French department who is also a student at GSAS, enrolled in the last three courses of her Master's Degree. NYUHome can tailor the channels she sees so that she receives information about staff meetings in the French Department, news from the GSAS dean, and commencement ticket information. This "butler" approach to information delivery can filter out unwanted information, eliminate junk mail and save time for NYUHome users.

The initial version of NYUHome, launched in June, contains 19 channels. It includes a mixture of channels based on academics, general information, personal information and entertainment. Students, faculty and staff were consulted to help determine which channels to include in the final version offered. To be useful, NYUHome needed to have information covering a wide variety of subjects, from academic reference and current headlines to entertainment.

NYUHome is competing with other portals, like MyYahoo and MyNetscape, and therefore we should not shy away from including popular features like horoscopes and movies. Future releases of NYUHome will continue to include an appropriate balance between NYU-related information and external feeds.

Work is already underway to add new channels to NYUHome. Because of the need for consistency, the NYU Office of Electronic Publications has been charged with creating and maintaining NYUHome channels. They can be reached at 992-9WEB.

Drew Hahn is the Director of Electronic Publications.
At the core of NYUHome, driving its different services and channels, is a portal engine. Developed by members of Information Technology Services, it was designed from the ground up to be a fast, highly available, secure and modular system. By creating our own engine, we had the freedom to build in the strengths of other popular portals, while adding many of our own innovations.

The front door to this engine is ISng, or “IS next-generation,” a tribute to the UNIX based IS menu/shell written by us for the old is* machines. This new endeavor was written using an entirely object-oriented framework.

Unique NYUHome Features

Before getting our hands wet designing NYUHome, our whole team looked closely at many other available portals. Most of them seemed based on the same mold, without much uniqueness or added value. In contrast, NYUHome stands out from the others because of several distinctive features:

Single Sign-on Pass-through Authentication

Once you log-in to NYUHome, you are one-click away from many university services. Our own session management mechanisms enable you to move from one server to another. One goal is that soon Albert will be modified so that a second authentication step will be unnecessary.

Surround Channel Technology

We designed NYUHome to be customizable. Users can easily change the look and feel of their NYUHome screen by moving channels around the screen based on priority or preference. All channels can be moved within the x-y grid. If a channel is moved off one side of an NYUHome screen, it will wrap around and appear on the other side of the screen.

MyHTML

We couldn’t find a single portal out there that was flexible enough to let users create channels with their own content, so we developed this feature for NYUHome. By giving users the ability to design their own channel, we can let them fetch their own data from external sources and display it for themselves. I have used my own MyHTML to keep pictures of my family, notes to myself, and other interesting messages. MyHTML is not a complex channel, but we consider it a prototype for the next generation of NYUHome.

Direct Messaging (DMT)

This is more similar to e-mail than to AOL’s Instant Messenger, but we built in this small internal messaging system because it was easy for us
to do. If you know a person’s NetID, you can send either text or HTML, and a Message channel will appear in the top-center of the recipient’s NYUHome screen, with a timestamp to show when it was sent. For security reasons, users should be able to restrict who can send them messages. Therefore, Direct Messaging can be configured with Allow and Deny lists.

**Built-in Information Gathering**

Writing our software from scratch has given us the freedom to build in our own mechanisms for incorporating information from both within and outside the University. Instead of using a complicated or non-standard programming interface (API), we have an array of very familiar tools and freely available perl modules, such as LWP and XML modules, for fetching information.

**ISng::Page**

*ISng::Page* is the core of our engine. It is what calls the methods to read in your configuration and calls the generator to build each channel you have chosen to display on your NYUHome screen. It is what provides our Surround Channel Technology, which lets channels move and wrap around in the x-y plane.

The entire HTML layout for NYUHome is programmed with simple tables, rather than with more complicated layers or java. Most of the core NYUHome system (*ISng* and associated channels) is not dependent on Javascript.

**ISng::Generator**

*ISng::Generator* is where all of the channels live. Each channel has the ability to read its own settings and generate its own content. If necessary, a channel can call the Fetcher to obtain information.

**ISng::Fetcher**

*ISng::Fetcher* is the data “grabber.” When called from the generator, it will obtain information from a remote source. For example, it might make a query to an Oracle database, or fetch XML based data stored remotely, or gather information from a news feed.

Because the generation and fetching process can take a long time, *ISng::Cache* runs between *ISng::Page* and *ISng::Generator* to keep load on our external data sources down and to keep pages loading quickly. Each channel in your configuration files has a setting called TTL, which stands for Time to Live. TTL dictates how long the channel data can be displayed without regeneration.

When a channel is generated for the first time, its output is stored in a file. The file is consulted each time your NYUHome page reloads. If the TTL has passed, the generator is called again and the cache is refreshed. Without this mechanism, users would have to wait a long time each time their screens load. Keeping the data in a flat file makes loading pages with fully cached channels extremely fast.

There are two cache stores, “Shared” and “Non-shared.” Channels such as Events, News and Sports use a shared cache. Channels that have private data, such as the Forums and Classes channels, use a non-shared cache.

There was lengthy debate within the development group about whether caching was really necessary for every channel, as we were modeling ourselves after TCP/IP and DNS caching mechanisms. While perhaps shortsighted, the motivation for this was good — we did not want users to get stale data.

By designing channels with the correct TTLs, this problem disappears. For example, the data collected for the Events channel changes once every 24 hours. We’ve set the TTL on that channel for 12 hours. In this way, there are only two queries made to our database per day, whether we have five users or five thousand.

The only time this architecture might have a problem is if we have two users, and the first user logs in at 11:59 p.m. In that case the cache would not be updated again until 11:59 a.m. the next day, and for the whole morning, the previous day’s calendar data would be displayed. With thousands of users logging in at all hours, the chance of this happening is very slim. The developers feel that this approach is much better than traditional caching.

Preferences are for fine-tuning the look and feel of the NYUHome environment. Your list of selected channels is compared to the list of all those available on the system, as well as against a list of access controls.
The Attributes option defines a style sheet that sets the color scheme of your NYUHome page. Users can also set their zodiac signs or preferred search engine. Channels such as Weather and Portfolios, which are coming soon, will have a rich set of options to define. This will also be the central point where your account settings, such as vacation message, forwarding and preferred e-mail address, can be set.

NYU Session Management System, or NYUSMS, is a separate but important part of NYUHome. It is a server written by Randy Wright that keeps track of users' sessions. When a user logs in, a session is created and a timer is started. Each time the session is validated, the timer is reset and the user is permitted to continue moving throughout NYUHome.

We felt the need to write our own session management system because we did not feel that any of the session management technologies out there are fast enough and flexible enough for our purposes.

NYUHome resides on “C1,” a cluster between a 8-CPU/8 Gb memory domain on our E10k, and a 4-CPU/4 Gb memory E3500. The web server for the application is Apache 1.3 with mod_perl. Mod_perl is an Apache module that builds a perl interpreter into the server.

In addition, we’re utilizing Apache::Registry, a part of mod_perl, that precompiles and caches the perl objects used for NYUHome services. This is an important feature in the NYUHome software architecture, because preloading the perl interpreter and precompiling the code reduces the amount of overhead to the server significantly.

In contrast, in an early development version of NYUHome, we did some benchmarking to demonstrate how many users we could sustain while serving logins within five seconds. Without mod_perl, we could handle two users logging in with a minimum load time of three seconds. After eight concurrent users, the load on the machine skyrocketed to somewhere in the range of 150 and the website was unusable.

Essentially, under our thresholds, we would have needed one CPU per concurrent user. We looked at the positive side and said that we only needed a few more E10k system boards to meet our goals.

With mod_perl, and a proper Apache configuration on the production environment, we can support more than 500 concurrent logins. After eight concurrent users, the load on the machine skyrocketed to somewhere in the range of 150 and the website was unusable.

With mod_perl, and a proper Apache configuration on the production environment, we can support more than 500 concurrent logins (that is, 500 users hitting the login button at the same time). Randy, who was doing the testing, could not open more than 500 sockets at once with his load tester, but extrapolating the curve, we can accommodate at least 700, maybe even 1000 people logging in at once.

Users will get their page loaded in under two seconds. This time does not take into account the amount of time your browser needs to render the HTML of your homepage.

When we unveiled the development version with mod_perl, the early testers on the development team thought Surround Channel Technology was not working. In fact, the page loaded so fast, they didn’t see the channel move! We estimate that based on this statistic, we can comfortably support at least 10,000 users logged in at once.

Future Work

Preferences and User settings will evolve greatly in the coming months. We are currently preparing to reorganize our LDAP (lightweight directory access protocol) directory structure to more closely match the hierarchy of the University. By doing so, we will be able to implement role-based access controls for Channels. We considered implementing a very simple system for this, so that when staff members log in, they won’t see the Albert channel unless they are enrolled in a class. Since we consider role-based ACLs to be critical to NYUHome’s success, we decided to hold off for the more robust approach.

NYUHome has been an ambitious project brewing for more than three years. It is remarkable for us developers to see it grow and mature to the point that it serves thousands of users every day. I look forward to seeing NYUHome continue to grow in the coming months and years.

Connect: Information Technology at NYU
NYU HOME Shared Services Environment
The NYUHome Shared Services Environment is the most powerful and capacious computing infrastructure at NYU.

It currently consists of a central machine, a Sun E10000 computer, with nodes labeled d1, d2, d3, d4 and d5 in the accompanying diagram. This machine has a total of 32 processors and 32 Gb of memory. The nodes act as separate machines. You log in to Home on the node d1, which is also where your files are stored. The e-mail node is d2. An Oracle database and other software associated with NYUHome are housed on d3, and d4 and d5 are used for beta testing.

The other five machines, labeled e1 through e5, are each Sun E3500 computers with four processors and four Gb memory. The machines e1 and e3 are mail gateway machines. The e-mail node is d2. An Oracle database and other software associated with NYUHome are housed on d3, and d4 and d5 are used for beta testing.

The NYUHome Shared Services Environment is the most powerful and capacious computing infrastructure at NYU.

In addition to their primary functions, e1, e3 and e5 act as failover machines for d1, d2 and d3. That means in the event of a system failure on d1, for example, the services on that machine will be taken over by e1 (see arrow marked "Failover"). This is possible because d1 and e1 are each dually connected to one of the one-terabyte (equal to 1000 Gb) disk arrays. This way, the data stored on the disk array are available to the failover node (e1) in the case of a failure of d1.

The www.nyu.edu cluster is on e2 and e4, and e5 is devoted to instructional management software — currently Blackboard.

David Ackerman
david.ackerman@nyu.edu

Connect: Information Technology at NYU

Fall 2000
NYUHome: A System of Systems

Gary Chapman
gary.chapman@nyu.edu

NYUHome is a portal, a gateway to a wide range of information services available from systems here at NYU and across the Internet. In this sense it is a system of systems, since NYUHome not only offers various services itself, but also pulls in information from other sources for immediate display. It can also lead us directly to other useful web-based services.

Behind the scenes, however, NYUHome is itself a very complex system of intertwined software components running on a number of computer systems and interconnected over a very complex internal network.

Hardware
The core NYUHome hardware is comprised of:
- One Sun Enterprise 10000 Server, configured into five independent computer systems,
- Five Sun Enterprise 3500 Servers,
- Three Sun disk arrays, containing 3 terabytes of fault-tolerant disk storage,
- Two Sun workstation console stations, and
- One APC Silicon Uninterruptible Power Supply.

Closely affiliated are a number of Sun and Compaq/Digital Unix systems, which provide a variety of basic services:
- Two Kerberos authentication servers,
- One Oracle database server,
- One Intranet server,
- Two directory servers,
- One backup server, and
- One Virtual Console system.

All this hardware is interconnected by a high-speed private data network, consisting of gigabit and 100 megabit-per-second switched Ethernet links. The whole assemblage is connected to the rest of the world by one of our core NYU-NET routers.

Software
On the software side, we find the following main elements:
- Sun Solaris Operating System,
- Sun High-Availability Cluster,
- Digital Unix Operating System,
- Apache Web Server,
- iPlanet Directory Server,
- iPlanet Messaging Server with Messenger Express mail client,
- Innosoft Directory Server,
- MIT Kerberos Distribution 5,
- Oracle Enterprise RDBMS,
- Sun NetBackup,
- Sendmail,
- File Manager,
- Lyris,
- Blackboard,
- Session Manager System (developed at NYU), and
- Iseng, Core NYUHome web-based software (developed at NYU, see Jonathan Vafai's article on page 7).

To see how these elements are at play when you use NYUHome, we can review what happens in a hypothetical first-time use of NYUHome.
Having connected to http://home.nyu.edu, you go through a first-time login process and agree to a statement of responsibilities. This is presented by the NYU-developed ISng application software, and delivered by the Apache Web server on home.nyu.edu.

How does NYUHome recognize that your NetID and password belong to someone eligible for an account, and how does it know that it is the first time you've tried to gain access? Basic information about each person eligible for an NYUHome account is stored in a directory database, derived from the central eServices Oracle database. This database contains basic information about the affiliation of each student and staff member at the University, derived from other administrative systems on campus.

Through automated processes, the NYUHome directory is updated on a continual basis with information about newcomers to the University. So when you connect to NYUHome, the directory is consulted to see if you are eligible to use NYUHome and if you have already activated your account. Then our Kerberos authentication servers are accessed to confirm that you have entered your password correctly.

Following your agreement to the statement of responsibilities, an e-mail message is sent to you ("Welcome to NYUHome!") and deposited in your Inbox, thanks to the iPlanet Messaging Server. In addition, a basic configuration file is created in your home directory, which remembers how you've customized your NYUHome screen.

If you select the Preferences option, the portion of the ISng software governing NYUHome channels is run, and you are offered a choice of channels to display on your screen. When you make changes to these preferences, your configuration file is updated; each time the main NYUHome screen is presented, your channel configuration file is consulted so that you see what you chose, positioned on the screen as you designed it.

Which channels are available to you? That information is stored in an Oracle database running on one of the NYUHome computer systems, and is updated as new channels are developed. In time, new channels will magically appear for you, since this Oracle database is consulted whenever you log in.

If you are subscribed to Lyris forums (NYU mailing lists), the Forums channel lists your forums, and allows you to click on any Forum to access it immediately without having to re-enter your NetID and password. Note that NYUHome and Forums are services running on distinct computers. It was a major design goal for NYUHome to enable communications between separate services so that re-authentication would not be necessary.

Behind the scenes, when you select the Forum you wish to enter, information is conveyed from NYUHome to the Forums server to confirm your identity and give you direct access.

A similar process occurs if you read your e-mail through Messenger Express, the built-in electronic mail client offered on the main NYUHome screen. Ordinarily, accessing a mail server that runs on a different computer from the main NYUHome interface would require you to re-enter your NetID and password. But fortunately, the iPlanet Messaging Server permits NYUHome, which has already confirmed your identity, to authenticate for you, allowing you to read your e-mail.

When you connect, the mail server consults the directory database (the same one used when you initially logged in) to get e-mail preferences and limits, such as the maximum amount of e-mail you can save on the system. So the same directory database is used both to store your basic NYUHome account information and the details of your e-mail configuration.

Perhaps you want to look up my e-mail address and send me a message. You can do this in Messenger Express, which has directory access capability built into it. Or you could use the Directory channel on the main NYUHome screen to get additional information, such as my office address and phone number.

These directory queries use the NYU public directory, technically known as ldap.nyu.edu, which can be accessed by anyone in the
world. This directory runs on two different computers, and only contains information about people and programs at NYU that is appropriate for wide public access. For example, if students do not wish to have their names and e-mail addresses published in the public directory, they will not appear there. However, they must still be listed in the private NYUHome directory database for their NYUHome accounts to work!

The Directory channel is an example of locally written software, designed to access ldap.nyu.edu and present the results in a carefully designed way.

By the way, you can send me e-mail using either gary.chapman@nyu.edu or gwcl@nyu.edu. The latter is my NYUHome e-mail address. Why do both work? The eServices Oracle database is the master repository of e-mail aliases such as gary.chapman@nyu.edu, created and modified through a web interface to the database. All such aliases are automatically propagated on a regular basis to smtp.nyu.edu, our central e-mail gateway server. (Incidentally, the aliases also propagate to the NYUHome directory database, so that the mail server can know the acceptable and preferred e-mail addresses for each individual.)

When a message arrives for gary.chapman@nyu.edu, one of the mail gateway servers determines where to direct it by consulting the local table of addresses derived from the Oracle database. A possible fancy future development would be to eliminate the automated propagation and local processing of such aliases in favor of direct lookups by the mail gateways, to determine how to route e-mail messages.

What happens if a hardware component within one of the NYUHome computers fails? For example, what if the primary home.nyu.edu machine has a malfunction? Since we don’t want home.nyu.edu to be unavailable, a major design effort went into implementing redundancy and reliability features.

A major component is the Sun High Availability Cluster software. Using this software, if home.nyu.edu fails, within approximately one minute, one of the Enterprise 3500 servers would automatically assume the role of home.nyu.edu, taking over for the failed machine. This server is able to access the same disk drives as the primary machine. It’s possible that we could have a failure and many NYUHome users would not even notice that they are suddenly using a different computer from the one they originally connected to.

After sending me e-mail, you may take a glance at the News channel. The news headlines may have changed since you last looked. Behind the scenes, the NYUHome system has consulted the remote system where the headlines originate. It does this periodically, and uses a sophisticated information-caching scheme to hold this information for display as people log in.

If NYUHome had to check the remote headlines each time a person logged in, it would be hopelessly slow, and the same information would be retrieved over and over until the remote server was itself updated at its source. This information retrieval, storage and presentation scheme was another major design and development effort that went into NYUHome. It was calculated to achieve the equally important goals of timely information and fast performance.

NYUHome’s goal is to provide ready access and a common interface for the NYU community to topics of widely shared interest, and to make integration of these services as seamless as possible. Future versions of NYUHome will link as tightly as possible to existing on-campus systems while adding new affiliated services, enhancing utility and overall ease of use. The background coordination of many internal systems and services form a strong foundation on which this evolution can take place.
New Uptown Connections
Wireless Networking at the IFA

Yoni Radzin
yoni.radzin@nyu.edu

The idea of installing cutting-edge wireless networking technology in an Institution dedicated to the study of art history that's located in a 18th century-styled building may sound a bit incongruous at first.

However, the installation and use of wireless networking at the Institute of Fine Arts proved to be appropriate, and above all effective. The Institute, one of NYU's far outposts, is situated in two separate locations: the James B. Duke Mansion, at the corner of Fifth Avenue and East 78th Street, and the Stephen Chan House, located diagonally across the street at 14 East 78th Street.

Before the implementation of wireless technology, only the Duke Mansion, an early 20th century building modeled on an 18th century French hotel, leased a high-speed T-1 line to NYU-NET. The Chan House, however, depended upon a vastly slower dial-up modem connection to the rest of the University.

To provide comparable network connectivity to faculty, staff and students in the Chan House, ITS' Network Services Group recommended the setup of a wireless Point-to-Point bridge, which would effectively connect Chan to the existing T-1 link in the Duke House. This setup would provide users in the Chan house (where the Institute's Conservation program is based) with more than adequate bandwidth to check e-mail, connect to various resources on NYU-NET, and search the Internet. In addition to being a proven building-to-building bridging technique (see the Fall 1998 issue of Connect, pages 11-12), the use of wireless technology, as opposed to older hardwired methods, ensured minimal alterations to an official landmarked structure.

Network Services employed Lucent Technology's Orinoco (formerly WaveLan) series of wireless networking gear to facilitate the wireless building-to-building bridge and provide adequate coverage throughout the six-story Chan house. The Orinoco equipment, like other popular wireless networking products (such as Cisco's Aironet and 3Com's AirConnect series), operates according to the IEEE 802.11b standard, meaning that the wireless devices transmit and receive data using the 2.4 GHz band. Currently, this allows for wireless data transmissions of up to 11Mbps, with fall-back rates of 5.5Mbps, 2.4 Mbps and 1Mbps (impressive, especially when compared to a typical 56Kbps and slower modem connection).

Orinoco products utilized at the IFA include WavePoint II access points, omni-directional and directional antennae, wireless PC cards, and Ethernet Converters. This series of products afforded the Institute with a cost-effective...
and flexible networking solution that is also scalable if the need for connectivity increases in the future.

The Lucent WavePoints, categorized as wireless transceivers, connect directly to the hardwired network (or a wireless skeleton), through a built-in Ethernet port. In turn, the WavePoints provide properly equipped client devices with access to the wired network. All client devices use an Orinoco PC card, a network interface card with an integrated antenna that facilitates radio communication with an access point. This specially designed network card can either be directly installed into the user's laptop or housed in an Ethernet Converter box, which then plugs into the user's desktop computer using standard twisted pair cabling (the desktop must have an available Ethernet port).

All elements of the Orinoco system utilize the wireless PCMCIA cards. In fact, the WavePoints can hold two PC cards at once, increasing the amount of network traffic they can carry (which is useful for load balancing). This robustness is characteristic of all of the Lucent Orinoco devices: the PC cards can communicate with multiple access points simultaneously, thus maximizing bandwidth and connection speed. The majority of the PC cards used at the Institute are Orinoco Bronze cards, which have a data rate of 2.4 Mbps.

Following a number of site surveys, in which Network Services determined the optimal placement of the bridge uplink in the Duke House and the receiving end in Chan, the first stage of the installation began in the fall of 1999. Immediately, astute observers at the IFA realized that the setup of a wireless network actually required some wires! In fact, this first stage involved the creation of wireless skeletons in the Duke and Chan houses.

In the Duke House, the bridge antenna, which was mounted in a carefully selected location in one of building's attic skylights, had to be wired into the Duke House's existing network. As for Chan, both the bridge and each WavePoint required a data jack to link to a 10/100 Cisco Switch that would carry the uplink from the Duke House.

"Wireless" networking was a little bit of a misnomer, as it turned out. However, the overall reduction of wiring, which simplified construction and reduced both time and cost, proved impressive. For example, only one wire run, to a single WavePoint, was required to provide network connectivity to all four offices in the first floor of the Chan house. Each office was equipped with an Ethernet Converter and wireless PC card to seamlessly connect to the local WavePoint, and in turn the network at large.

As the wireless project at the IFA evolved and installation began, the scope of the project gradually expanded to include areas of the Duke Mansion, which required network connectivity, but had not yet been wired for data.
Utilizing the Orinoco technology to provide high-speed network access in locations like the third floor of the Duke House proved to be extremely efficient and cost effective. Instead of creating 18 separate hardwired jacks in each faculty office on the third floor, only two data jacks were necessary — one for each of the Wave Points on the third floor.

An earlier site survey had established that two WavePoints placed on either end of the floor could provide adequate coverage to all offices on the floor. Again, the WavePoints were mounted approximately 10 feet above ground to limit signal interference and avoid any physical obstruction.

Once on line, the wireless networking gear performed as promised. Users that had reluctantly become accustomed to dial-up modem connections could now quickly and easily access network resources. Not surprisingly, considering the recent development of the technology, a few minor glitches surfaced. However, Network Services and Lucent quickly provided solutions.

Although stable the majority of the time, the Chan-Duke connection occasionally timed out, resulting in loss of network connectivity in the Chan house. A temporary remedy for the problem, re-booting the two bridge/WavePoints, quickly became tiresome. Soon after the problem was detected, the main culprit was identified — the Duke bridge’s omni-directional antenna. When replaced with a directional antenna aimed directly at the Chan bridge, the wireless link stabilized and even increased in strength. Other more minor problems, including the inability of the PC cards to use AppleTalk, were quickly remedied by firmware upgrades provided by the vendor.

The wireless networking project at the IFA did not end when the Duke-Chan bridge first came online this past December. The effectiveness of the Lucent solution, notably the ease of set up relative to hardwiring and the scalability of the Orinoco products, persuaded the Institute to install WavePoints in additional locations in the Duke Mansion. The set up of a WavePoint in the second floor balcony area, for example, provided connectivity to several remote faculty offices that required high-speed Internet connections, without involving expensive or disruptive construction.

In addition, the effect of the construction on the landmark building’s appearance and infrastructure was minimal and in most cases not perceptible. As of this writing, WavePoints located in the Chan and Duke houses at the Institute can potentially provide coverage for most populated locations in each building. Easily upgradable, through firmware upgrades or PC card swapping, the system should provide the Institute with fast network connectivity far into the future.
Collecting Data Over the Web
Data Entry for SPSS

Frank LoPresti
frank.lopresti@nyu.edu

The Statistics Group of ITS Academic Computing Services has acquired SPSS Data Entry software for data collection over the Web. The data entry tools in this package will allow researchers at NYU to build forms and questionnaires, to deploy them on web pages, and to collect the data entered by respondents. The three Data Entry tools are Builder, Web Server and Network Server.

Designing a Questionnaire
Start with the Data Entry Builder to design a questionnaire. Data Entry Builder includes everything you need to build custom forms, enter data, and check data for accuracy using rules. Builder offers an easy-to-use graphical environment for defining variables and questions, developing custom forms, and creating Validation, Skip-and-Fill, and Checking rules to ensure clean data. Sample surveys and question libraries are also included. If you’re a lone researcher or a one-PC shop, a single copy of Data Entry Builder should have everything you need.

The editing tools in Builder let you place variables by dragging them from a variable window. The variable list may be built, and the variables defined, named and formatted, during the editing session. Alternately, variable names and definitions from an existing data set in SPSS or Excel can be opened in Builder to create a list. Another window, similar to the variable list, stores the questions associated with the variables. These variables and questions are saved in files and are edited as the questionnaire is being built. They can be saved and used later in a new questionnaire.

Answers can be collected with check boxes, multiple choice, pull-down lists or text input. You can also set rules, such as Skip-and-Fill, to control the route through the questionnaire.

SPSS provides many sample surveys to help you get started, copy preformatted questions from question libraries, or build your own forms from scratch. You can also generate forms from existing SPSS data files.

Creating Paper and Online Forms
You may find that you can use the same form in print that you use online. Or you can develop alternate versions of a form — one optimized for online data entry, the other used for your printed questionnaire. You can map questions on both forms to a common set of variables so that if you make changes to one of your variables later (for example, to the wording of a question), questions on both forms are updated automatically.

Successful surveys begin with well-defined questions. Too often, researchers consider how to enter and analyze their data only after they’ve mailed their questionnaires. In Data Entry, as you create your questions and responses, you also define the variables you’ll use to conduct your analyses. If you start with an SPSS file to generate your survey, this result is guaranteed. Either way, the result is clean data, ready to analyze.

Fast, Efficient Data Entry
Data Entry has a number of features to speed online data entry. You can create Skip-and-Fill rules to automatically fill in
specific questions based on previous responses, or use the Table Entry view to quickly enter data into a tabular grid similar to a spreadsheet.

**Clean Data Entry**

Create Validation rules to ensure that data are entered correctly the first time, or use Checking rules to screen for logical inconsistencies (such as a male respondent who claims to be pregnant). You can also check for errors in your existing data files. Use the file compare facility for double-entry verification.

Analyze your data using SPSS's statistical software. SPSS Data Entry is designed to be used with statistical software from SPSS Inc., maker of powerful tools for data analysis. You can read your Data Entry files into and out of SPSS programs at will. Throughout the documentation, tips and techniques are described to help you get the maximum benefit from both Data Entry and your data analysis software.

**Deploying Surveys on the Web**

Use Data Entry Web Server to export the completed survey as html. Then, use the Network Server to register it on the server.

**Collecting Data**

SPSS Data Entry Network Server collects data in an SPSS format. You can bring your Data Entry files into and out of SPSS programs at will, and analyze your data using statistical software from SPSS.

Collecting data from a web page is clearly useful for modern research. In the Spring 2000 semester, a professor doing research on web usage used our software to design a trial questionnaire in the Statistics Group Innovation Lab in Tisch Hall. The survey was linked to selected commercial web pages. The respondents would be web surfers. Users who randomly, serendipitously surfed across sites with the questionnaire link would be able to respond. Questions were asked pertaining to the personalities, habits and appetites that lead to the person being on these selected pages.

These Data Entry tools make it possible for the "non-programmer" to create and use attractive web surveys. These higher-level programming tools allow us to write programs for the Web and web servers without knowing web languages such as Perl and Java. Using pull-down menus to generate web applications is similar to programming statistics using SPSS. It's user-friendly and it works.

This software is installed on a web server at the Statistics Group Innovation Lab. Contact Frank LoPresti for more information.

---

**New Software Avoilable Through the Social Science Stats/GIS Group**

The ITS Tisch Hall Lab in room LC-8, and the Social Science Innovation Lab in room LC-7 provide SPSS version 10 and SPSS DE Web Builder. Both applications are described in short articles in this issue of Connect. Additionally, we now have the new SAS version 8 — a complete statistical programming language — and Version 8 of ArcInfo, our premier Geographic Information System.

S-PLUS has been available at the Tisch Hall computer lab for some time. New add-on modules for S-PLUS — spatial analysis and GARCH — are now available in the Social Science Innovation Lab.

S+GARCH, to quote the manual, "provides an essential suite of tools designed for the statistical analysis of financial time series data." Financial time series modeling is different from typical time series modeling because of conditional variance information in its models and predictions. GARCH is specifically designed to look at second order moments (variance), important to modeling risk. Evaluating time dependencies can give more accurate modeling and make better predictions.

**New Scanner Software to Read Questionnaires**

Now Remark Office OMR 5.0 lets you scan questionnaires to create a dataset. Image fields have been updated, and the software now automatically detects if data was handwritten into the image field. Whenever handwriting is detected, the cell in the template grid is colored blue. The technician need not open an empty cell if the respondent didn't enter text.

The software recognizes optical marks, such as bubbles or checkboxes, as well as barcodes. With Remark Office OMR, you can design your own forms using any word processor, print them on your printer, scan and recognize data with your image scanner and then export the data to the application of your choice, including SPSS.

A demonstration scanner and machine with this new software are available in the Social Science Innovation Lab. Call Frank LoPresti at 998-3398, or send e-mail to lopresti@nyu.edu, to arrange for a demonstration.
Version 10 of SPSS introduces a very important difference from older versions — a dramatic retooling of the Data Editor.

The changes to the SPSS Data Editor in this new version are significant. The spreadsheet — the entire view used when creating an SPSS data set — is different in version 10. However, all the concepts involved in creating a data set and defining variables are still valid. The mouse clicks and the new screens you see are also an improvement.

The new Data Editor provides two views of your data.

The Data View in Figure 1 displays the data in a spreadsheet format. Each column is a different variable and each row across the spreadsheet is a case. A case is usually the unit of study in our research, such as a respondent’s answers to our questionnaire.

The Variable View in Figure 2 is also a spreadsheet, but it lists our variables rather than data. Each row across names a variable, such as ID, AGE or SALARY. It then includes other information about the variable, such as each variable’s attributes. In older versions, each variable’s information was presented in a separate “Define Variables” window. Now, in Version 10, you see the information about all the variables at the same time.

So the old Define Variables window — which was used to allow us to define and change the variables’ names and to define the four attributes used when creating an SPSS data set — is gone! Now we look at a second spreadsheet, the Variable View, to view all the information we put into defining the variables.

In the Data Editor window, you move between the Data View and the Variable View by clicking on their tabs on the lower left in either view; they are pointed out in Figures 1 and 2.

Defining Variables in a New Data Set

Each row in the Variable View spreadsheet names and defines the attributes for each variable in your data set. The columns in the Variable View spreadsheet are assigned values that determine the attributes for each variable. The first column is used to name the variable being defined. The Name field is limited to eight characters without spaces.

After the Name column, the remaining nine columns of the Variable View are used to manage the attributes for that variable.

Click within the Type cell and an icon appears (see Figure 3). Click on that icon and a window opens where you select the type of variable (see Figure 4) such as “numeric,” “date” or “string.”

Choose the length for the variable with the Width column.

Use Decimals to change the number of significant integers for numeric variables.

Since SPSS only lets you use eight characters to name a variable, the Label field lets you elaborate on the name. MOTHERS_INCOME is a great name,
but it is too long. So you can name the variable M_INCOME, and then label it “Mother’s Income” in the Label field. This label is used in statistical procedure output. For example, if you make a table involving M_INCOME, the label “Mother’s Income” is used. That variable label becomes part of the data set. When you give the data set to someone else, these labels are included. The data set becomes self-documenting.

Value labels allow you to record the meaning of categorical values of a variable. For example, you might have a variable COB, Country of Birth, in which 1 stands for USA, 2 is Argentina, 3 is Brazil, and so on. These values can be recorded when you create the data set. These labels are also used in output presentations.

Incomplete data is a common problem. Analysts hardly ever get a complete data set. Rarely is every question answered on a questionnaire. But we can’t throw away an entire questionnaire if the person won’t answer one question out of a hundred. Using Missing Values allows us to deal with missing data.

Columns and Align control the display properties of a variable, such as the width of the column in the spreadsheet display.

**Creating a Sample Data Set**

To demonstrate variable definition, here are the steps to create a simple data set using information on age, gender, and salary.

1. **Define a variable named ID, which will serve as a unique identifier or key for our cases.**

   Be sure you are on the Variable View within the Data Editor window. Enter “id” in the first Name field. The default description usually describes a numeric variable, and therefore is probably suitable for the ID variable. Be sure there are no decimals in the definition.

2. **Define a variable named Age.**

   Enter “age” in the second row for the variable Name field. Again, the default description is probably okay.

3. **Define a variable named Gender as a numeric variable with value labels.**

   Enter “gender” in the third row Name field for the third variable.

   The variable name Gender is sufficiently descriptive, so we
will not add a variable Label. But we will add value labels.

Single-click on the cell in the Values column. An icon appears in the cell. Single-click on that icon and a Value Label window appears. Enter “0” in the Value field and “Male” in the Value Label field. Click the Add button. The phrase 0=“Male” appears in the lower section of the dialog box. Now, enter “1” in the Value field and “Female” in the Value Label field. Click on the Add button again.

We are using the values 0 and 1 to represent the two genders, but the choice of numbers is arbitrary. Numeric values are often assigned to a categorical variable when a questionnaire is designed. When that is the case, use the reassigned numbers to define value labels.

Click on the OK button. The Define Labels dialog box closes.

NOTE: Do not to click the OK button until all of your labels appear in the lower section of the dialog box. If you do not click on the Add button after entering a Value and a Label, the information will be lost.

Define a variable named Salary with a variable label “Salary in Thousands.”

Enter “salary” in the fourth variable Name field. In the Labels cell, type in “Salary in Thousands.” The fourth column now contains the labeled salary variable.

All of the variables in our sample data set are now defined. Save the data set and you are ready to enter data. Enter the data into the Data View spreadsheet one cell at a time. Again, toggle between the Data View and the Variable View using the tabs at the lower left of the SPSS Data Editor window.

Other functionality, outside the scope of this article, will quickly unfold to the experienced Windows user. For example, as you experiment with cutting and pasting these definitions within the Variable View, you will see how to create other variables with the same data definitions from one variable definition. We could always cut and paste within the data spreadsheet; now it is possible in the Variable View spreadsheet as well.

The new Data Editor for Version 10 offers much to the SPSS programmer. The Variable View allows us to see and manipulate data definitions for the entire data set in one window. Older versions only allowed us to inspect one variable definition at a time.

For more complete information on the new Data Editor, see the SPSS Base 10.0 User’s Guide, published by SPSS, Inc.

This article is a partial update to the NYU/ITS SPSS for Microsoft Windows, V9.0 document (12/99: www.nyu.edu/its/socsci/Docs/SPSSwin9.pdf). To this end, users of V10 should refer to these pages for information about the Data Editor. This article replaces pages 3 through 7 of the aforementioned document beginning with the section titled “Part 2. SPSS Basics, Creating a New Data Set, Defining Variables.”

Otherwise, the SPSS for Windows Version 9 document, (also available at the Tisch Hall, LC8 Academic Computing PC site) is still useful to the new SPSS user. The skills needed to perform data analysis in SPSS are accurately detailed in the Version 9 document.
Webcasting in the New Millennium

Robin Schanzenbach, Gloria Rohmann, José Calero and Toni Urbano
www.nyu.edu/nyutv/

Webcasting, also referred to as audio, video or media streaming, is a generic term for a technology that allows audiences to view television programming or listen to radio programming over the Internet. Webcasting is revolutionizing the television industry because it can be accessed at any time of the day, from any corner of the world, over the Internet. This year the webcasting medium celebrates its fifth birthday. The toddler stage, with all its stumbling with different standards and crawling for bandwidth, is at long last behind us. NYU Libraries’ Avery Fisher Center and TV & Media Services, in collaboration with ITS, have finally cut their teeth and successfully executed several live events supporting this technology. Together we continue to prepare ourselves for an unprecedented growth spurt and structural complexity that will transform the technology from a slow and jerky Internet novelty into a major new communications medium.

Bobst Library’s Avery Fisher Center and the Television Center first used streaming media in the fall of 1997, when we webcast the proceedings of a Media and Democracy Conference in live audio. We received the analog audio signal over a simple analog telephone line from Cooper Union’s Great Hall and plugged it into a server in Bobst’s Electronic Resource Center server room. The next big test of the technology came with Commencement 1998, when we successfully streamed both audio and video. The event was produced in the park in a television production truck and transmitted with microwaves from the truck to a receiver on top of the School of Education Building. Then the signal traveled over fiber optic cable to the Television Center, where it was modulated over the NYU Broadband and then demodulated at the ERC server room for encoding onto the Internet.

Having proven that the technology worked and the team effort was a success, we were ready to move on to a higher level of complexity, which would allow us to expand the audience. NYU Libraries and TV & Media Services have production resources and technical expertise that allow us to ramp up our webcasting services. Campus Media supports live events on campus; the Television Center has high-end video production and post-production tools; and NYU-TV serves as one of the primary vehicles for content delivery to the university community. Lectures, guest speakers and events are staples of NYU-TV programming. However, NYU-TV is normally distributed over
Campus Cable and the Broadband system, and cannot be accessed by faculty or students who live off-campus. Due to these constraints, webcasting is the next logical step in distributing NYU broadcast content. The developing NYU Digital Library includes archives and catalogs of these webcasts.

In the fall of 1998, the NYU School of Law’s “Strengthening Global Democracy” conference with President Bill Clinton and Prime Minister Tony Blair provided a great opportunity to work with this technology. We worked in conjunction with the White House and an outside vendor to provide the live encoding and large bandwidth required for such a historic event.

When making plans for Commencement 1999, we collaborated with ITS to expand our bandwidth and increase the licensing that would firmly lay the foundation for reliable webcasting services here on campus. Due to its broad appeal, Commencement is the perfect webcasting opportunity. Friends and family who are unable to attend the ceremony in person can log on and watch from anywhere around the world. The webcast was viewed in Japan, South America and several locations in Europe.

Anyone who has ever taken a film, music or language course, or any class requiring the use of media other than print, will certainly recall the inevitable trip to the library to view video or listen to audi-taped educational materials. These materials are an obvious choice for streaming media. In the summer of 1998 the Avery Fisher Center encoded language tapes for French, Spanish, Italian and Japanese in Real Audio and made them available for students over the same ERC server that fall. When ITS set up its industrial-strength server and software early in 1999, the AFC started experimenting with encoding music for class reserves. Fall 1999 saw the first roll-out of music files for Kent Underwood’s School of Education course, “The History of Western Music.”

We hope to be adding several new music classes in Fall 2000. All music selections are taken from recordings owned by the Avery Fisher Center. We
The synergy that webcasting creates between audio-video content, publicity and distribution makes it invaluable for our educational purposes. Our success with past events can serve as a model for delivering NYU conferences, lectures and events to the NYU community and even beyond, to those who can't otherwise attend due to space, location and time constraints. Unlike past content delivery systems that are confined to physical spaces and specific time slots, webcasting allows immediate and convenient access to education, information and entertainment material 24 hours a day, seven days a week. The digital library of the future will include sound and sight.

Although at present webcasting may not offer the same broadcasting quality as satellite or cable television, it has already surpassed these other technologies in affordability and ease of use. Webcasting and streaming media is on a parallel gestation track with the evolution of the Internet. Television is shedding its analog cocoon and quickly developing the digital wings that will allow us to continue servicing the video and audio demands of the University in the Internet age.

For further information on webcasting or NYU-TV, contact Jose Calero of the Television Center at 998-5168. To view archived materials, consult www.nyu.edu/nyutv/, and click on “online events.” For more information on digital reserves, call the Avery Fisher Center at 998-2534.

are using Real Audio, encoding the files at 20Kbps (providing acceptable sound over 56K modem dial-up connections). Streaming services for course reserves have been very popular (see www.nyue.library/bobst/e-reserves/). During the 1999-2000 academic year, there were 24,341 hits for language and music combined.

The basic architecture needed to disseminate or archive broadcasting material over the Internet is straightforward: a programming source such as a video or audio signal, an encoding station, and a media server. On the receiving end, the minimum requirements to view or hear a webcast are also relatively simple: a computer equipped with a 486 processor, 28.8 kbps modem, 16 megabytes of RAM, a sound card, speakers, and the appropriate software to play streaming media files. The Television Center formats its programming to be viewed using Real Player software, which can be downloaded for free. The TVC’s encoding station uses Real Producer G2 software to encode media files. Analog video (such as videotape) can be encoded, or video already digitized as MPEG, AVI, and Quicktime may be converted to RealVideo for streaming.

Webcast images piped over existing phone lines at 28 kilobits per second are often jerky and slow; allowing only 10 to 12 frames per second. Because broadcast television is delivered at 30 frames per second, this disparity becomes even more accentuated when viewing a program with a lot of movement, such as a sporting event, although it is acceptable for panel discussions and lectures. Soon, however, with the commercialization of broadband-type technology coupled with the exponential growth of the Internet’s audience, these shortcomings will be a thing of the past.
Welcome Home.

NYU Home
http://home.nyu.edu